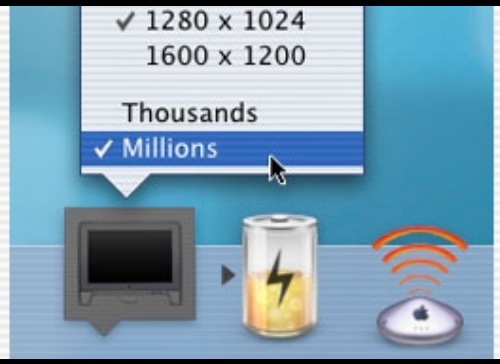
Session 601

WebObjects 5 Technical Overview



Steve Hayman
Systems Engineer
Apple iServices

Introduction

- Understand the WO architecture
- See the tools, frameworks and development process in action

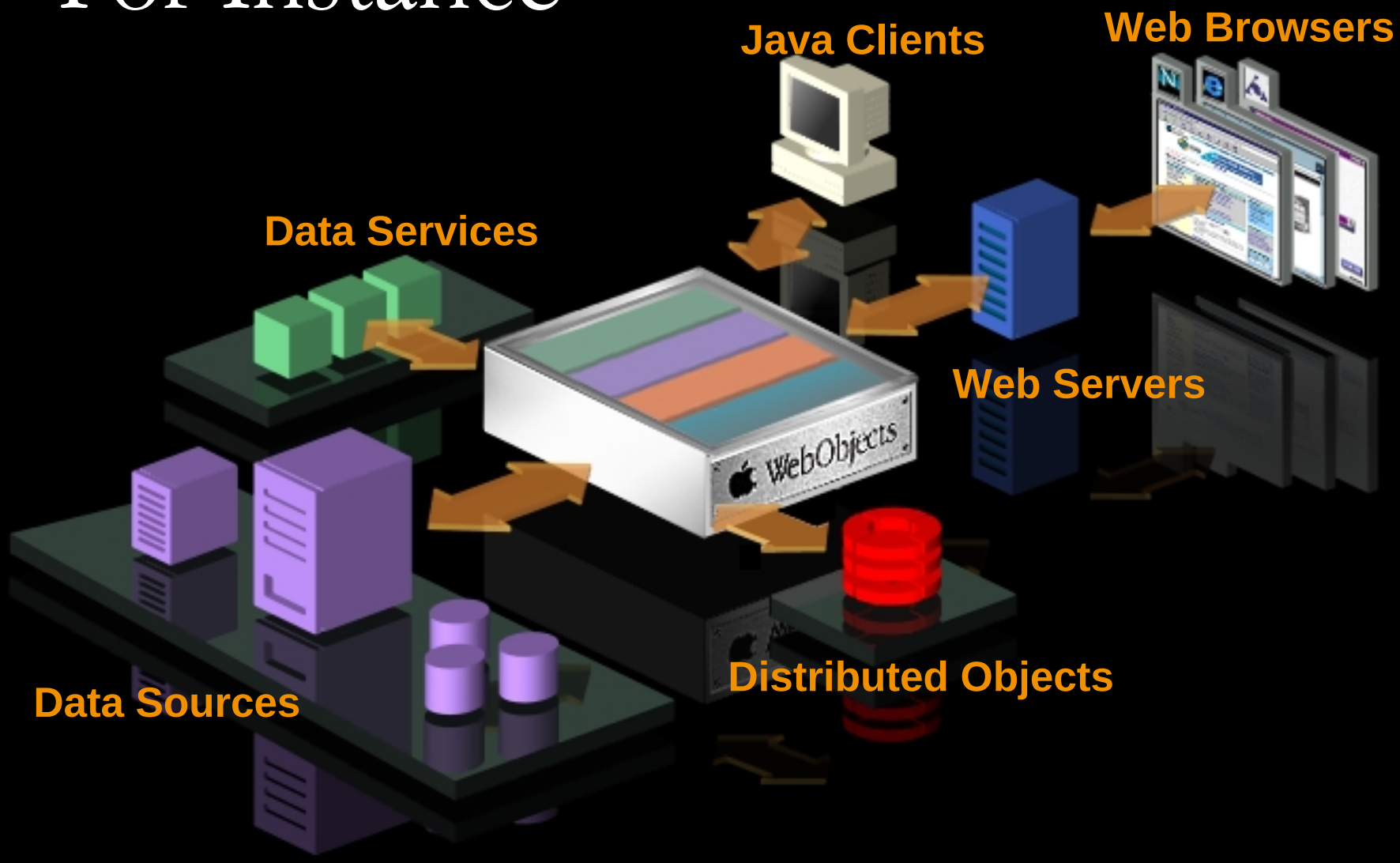
What You'll Learn

- WO Tools
- WO Frameworks
- WO Deployment
- Understand the titles of the other WebObjects sessions
- . . .And what is the deal with WebObjects 4.5.1?

What WebObjects Does

- It's an *Application Server*
- Who invented that term anyway?
- It's more than just that
  - Developer Tools
  - Pre-written objects (Frameworks) ← The best part!
  - Runtime environment
  - Monitoring and control tools

For Instance



Who WebObjects Is For

- Developers
- Very powerful
- Steep learning curve
- But that's OK

WebObjects Works With…

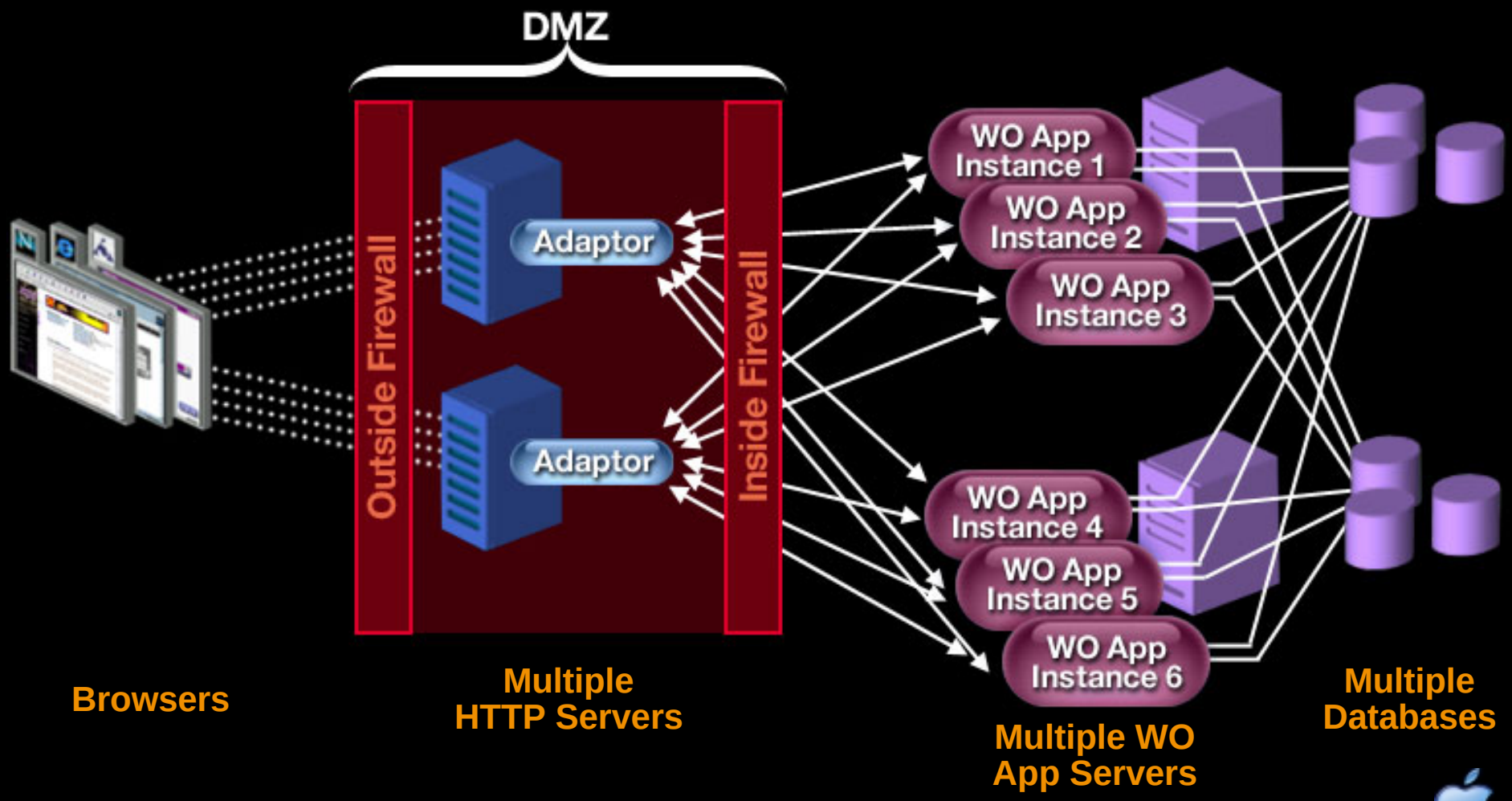
- Java
- Databases—JDBC ,Oracle, Informix, Sybase, SQL Server, LDAP, OpenBase, FrontBase…
- Different UIs—HTML, WAP, PDF, XML
- Most web servers—Apache, Netscape, IIS
- Lots of 3rd-party objects and APIs
- Open open open open open open open open

Where You Can Use WebObjects 5

- Develop on Mac OS X or Windows 2000
- Deploy on Mac OS X, 2000, Solaris, HP/UX, Linux
  - (Theoretically, anywhere with Java 2)
- Mix and match as needed
- Start small and grow bigger

Start With This…

. . .And Scale Easily to This



Key Ideas

- Web Components (front-end)
- Enterprise Objects (back-end)
- Kept completely separate
- Bound together at runtime

Web Components

HTML templates

+

<WEBOBJECT> tags that indicate where something interesting and dynamic happens

+

Bindings that fill in the blanks

+

Lots of pre-written objects to re-use

What Comes Out of a Component?

- A string—of your choice
  - Plain text
  - HTML
  - Image data
  - JavaScript commands
  - AppleScript commands
  - . . .

Non-HTML Components Too

- PDF
- WAP
- XML

Mix and match

Enterprise Objects

- Java objects that implement your business rules…
  - Customer, Order, Product, Vehicle, Package, Option
- JDBC Adaptor that speaks to databases
  - Oracle, Sybase, Informix, LDAP …
- Models that connect objects to databases
  - Customer ⟷ Oracle CUST table
  - Vehicle ⟷ Informix VHCL table
- Relationships between objects
  - Customer has a list of Vehicles

EOF Is the Real Gem Here!

- Pure business objects
- Totally UI independent
- Totally database independent
- Persistence is completely managed for you

. . .All Bound Together. . .

- WO app
  - Loads components
  - Finds **<WEBOBJECT>** tags
  - Fetches and sends messages to Enterprise Objects
  - Mixes results into the component
- User sees plain HTML (PDF, WAP, XML)
- Customize the entire process at runtime if needed

. . . With Advanced Session Management

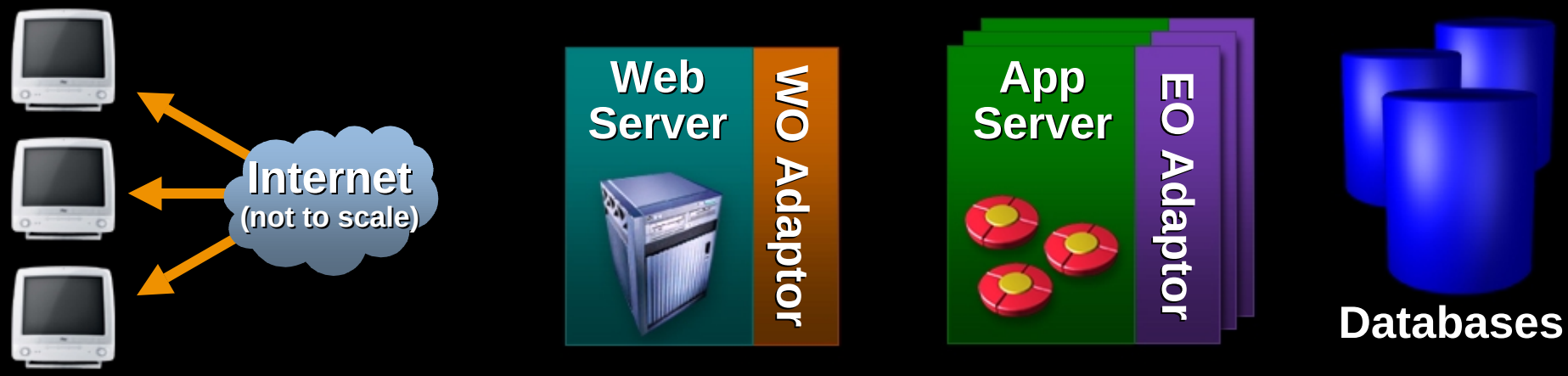
- WO manages an extensible Session object for each individual user
- Also an application object shared by all
- Complete control over session creation and archiving
- Extend Session to add security, shopping carts, user preferences, personalization and so on

. . .And Runtime Monitoring Tools

- Monitor
  - WO application that manages your big honkin' deployment environment

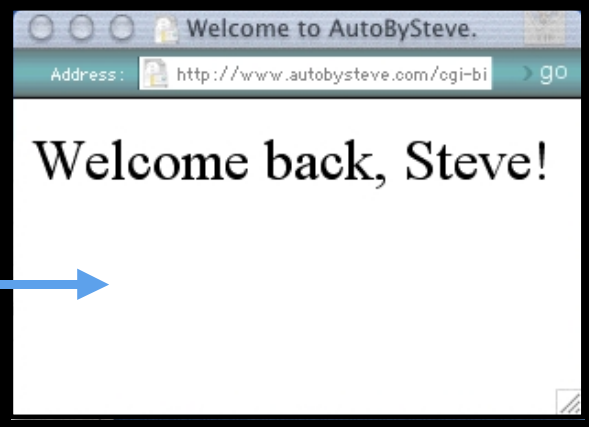
WebObjects in Action

Any OS Any Web Server	Apache Netscape IIS CGI	Mac OS X Windows 2000 Solaris HP/UX Linux Other Java 2 Platforms	JDBC Oracle Sybase Informix SQL Server LDAP OpenBase FrontBase ...

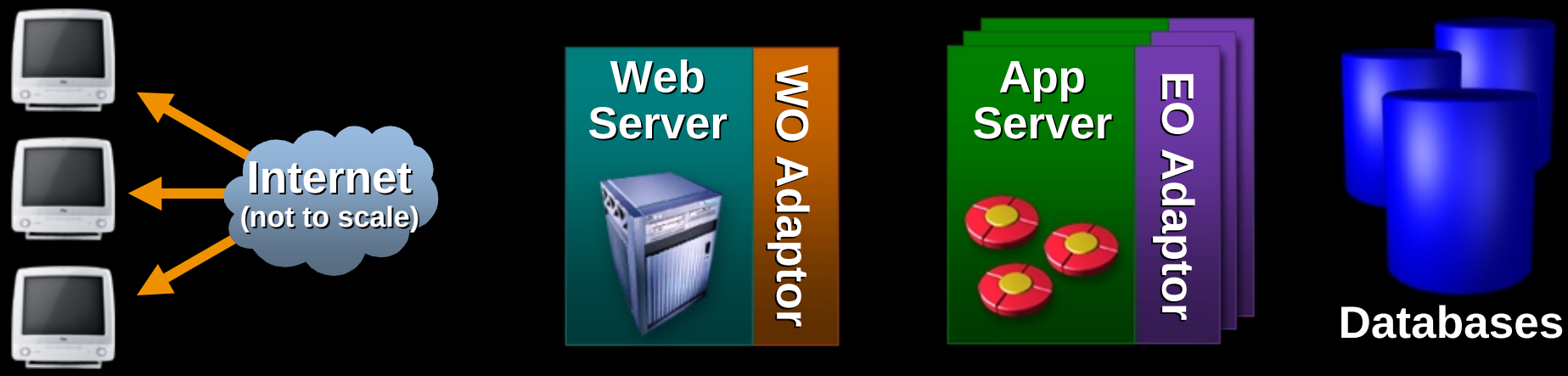


A really simple app

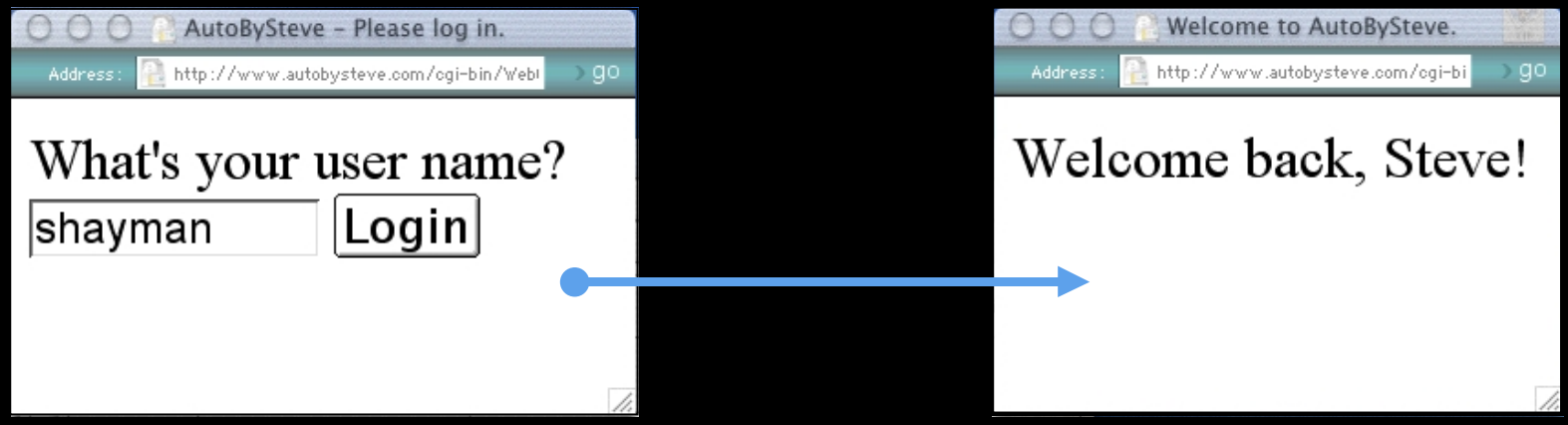
1. Process a request



2. Generate a response



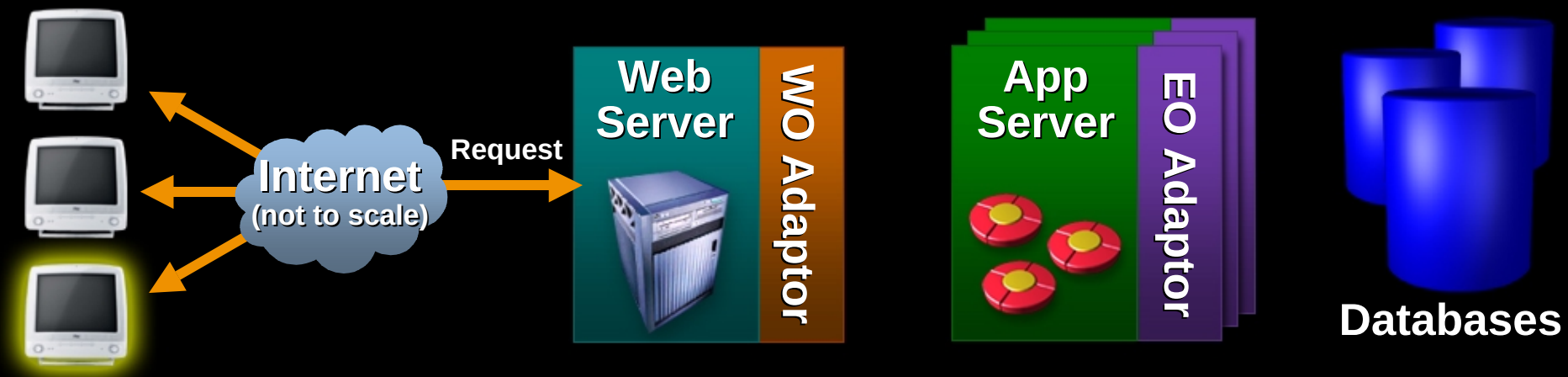
3 Steps WO does for you automatically



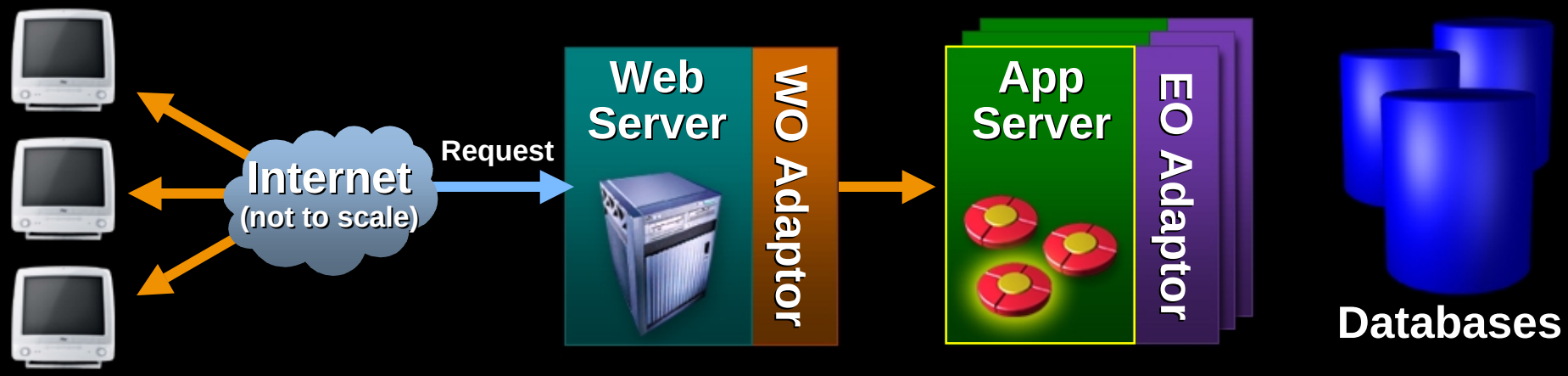
1. Take values from request

2. Invoke an action

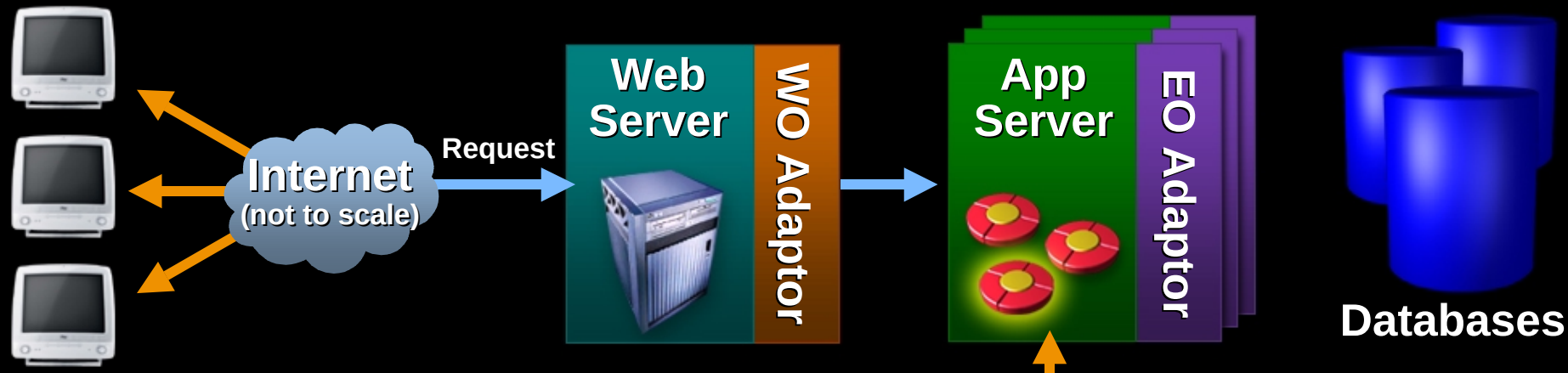
3. Generate a response



- User submits request to web server



- Web server adaptor finds appropriate application server and instance, and forwards the request to it
  - First request? A random instance is chosen, and a new Session created
  - Subsequently?
    Return to the original one

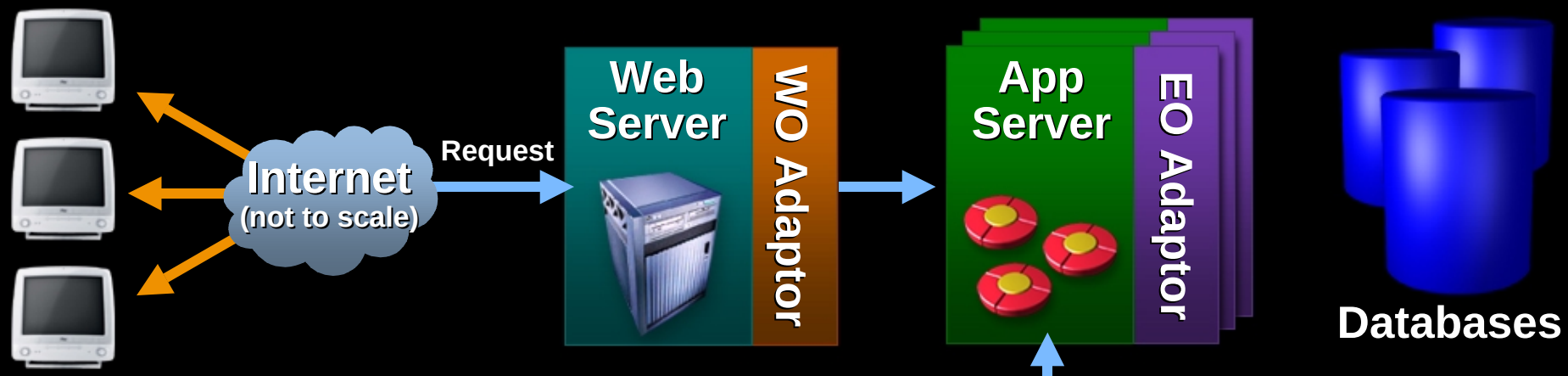


- Session finds the requesting WOComponent

Login.wo

```
<HTML>
What's your user name?
<WO NAME=FORM>
<WO NAME=INPUT>
<WO NAME=SUBMIT>
```

```
INPUT: WOTextField
{ value = userName; }
SUBMIT: WOSubmitButton {
  action = handleLogin; }
```

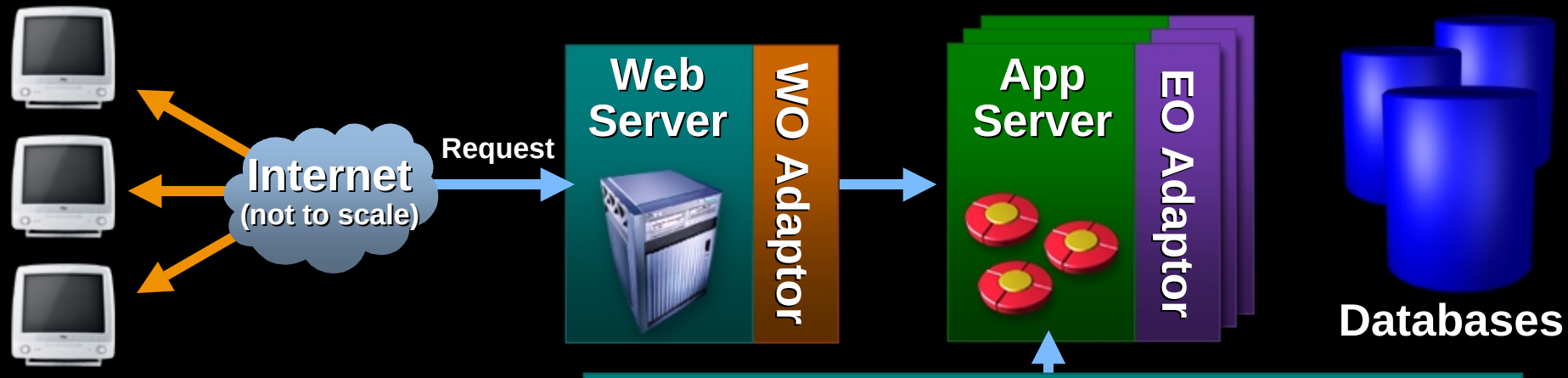


- The requesting component studies any form values that have been submitted

Login.wo

```
<HTML>
What's your user name?
<WO NAME=FORM>
<WO NAME=INPUT>
<WO NAME=SUBMIT>
```

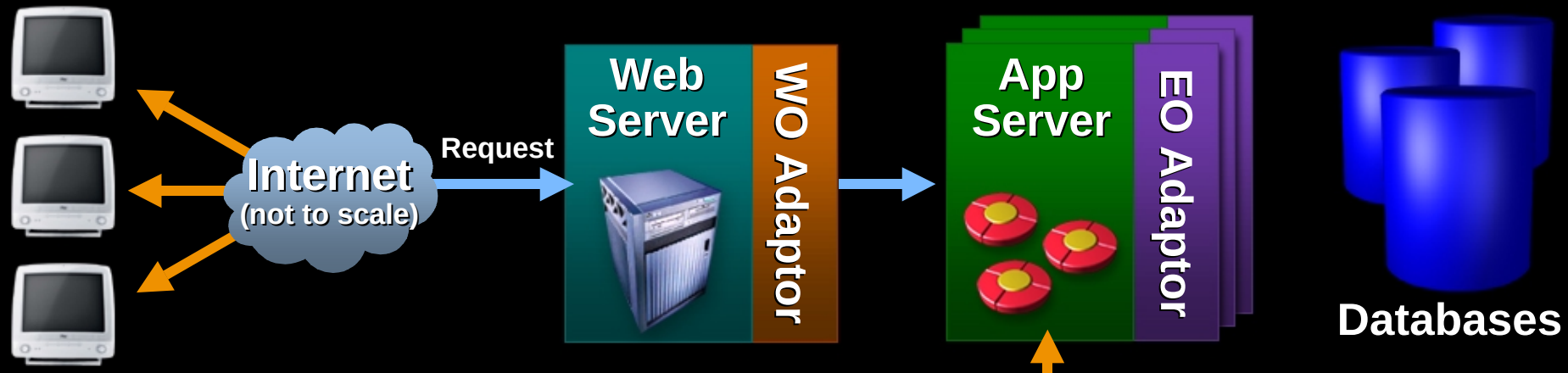
```
INPUT: WOTextField
{ value = userName; }
SUBMIT: WOSubmitButton {
  action = handleLogin; }
```



- Requesting component then figures out what action to invoke
  - Action returns another component

```
//  Login.java
public class Login extends WOComponent {
  public String userName;
  public WOComponent handleLogin() {
          session().fetchCustomer(userName);
          return pageWithName("Welcome");
  }
}
```

```
INPUT: WOString
{ value = userName; }
SUBMIT: WOSubmitButton {
  action = handleLogin; }
```

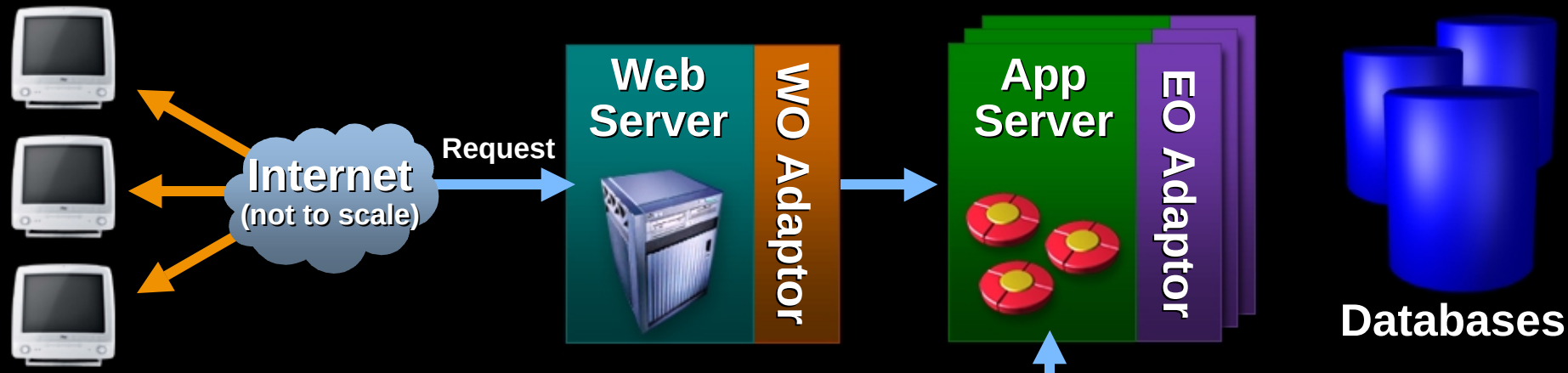


- Action is invoked, and session loads the resulting response component

Welcome.wo

```
<HTML>
Welcome back,
<WEBOBJECT
NAME=UserName>
```

```
UserName: WOString
{ value =
 session.customer.firstName;
}
```

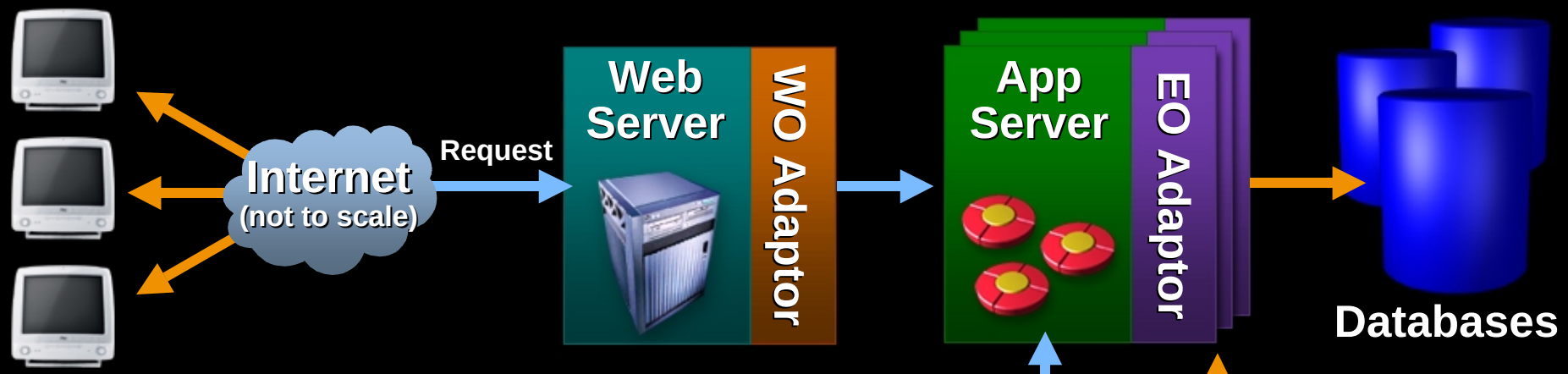


- WEBOBJECT tags and corresponding messages are identified

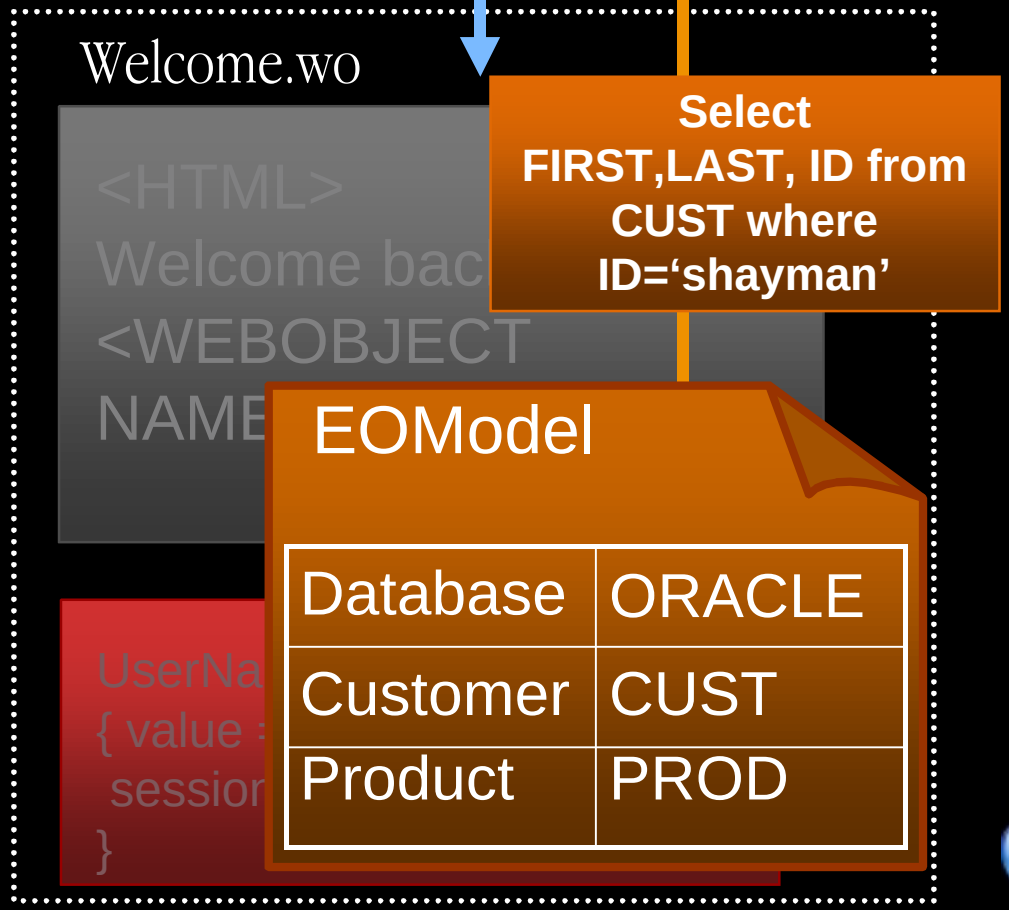
Welcome.wo

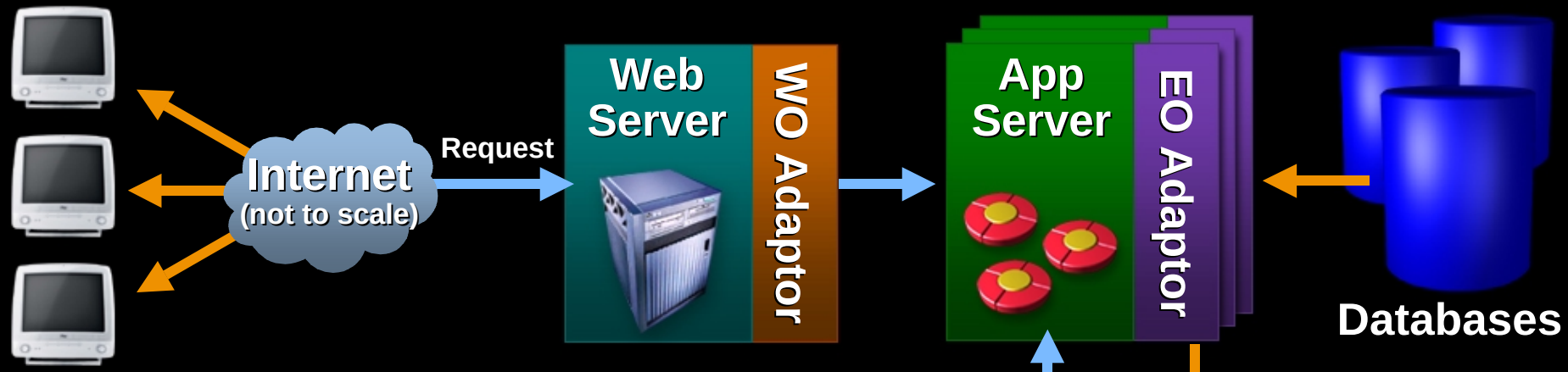
```
<HTML>
Welcome back,
<WEBOBJECT
NAME=UserName>
```

```
UserName: WOString
{ value =
 session.customer.firstName;
}
```

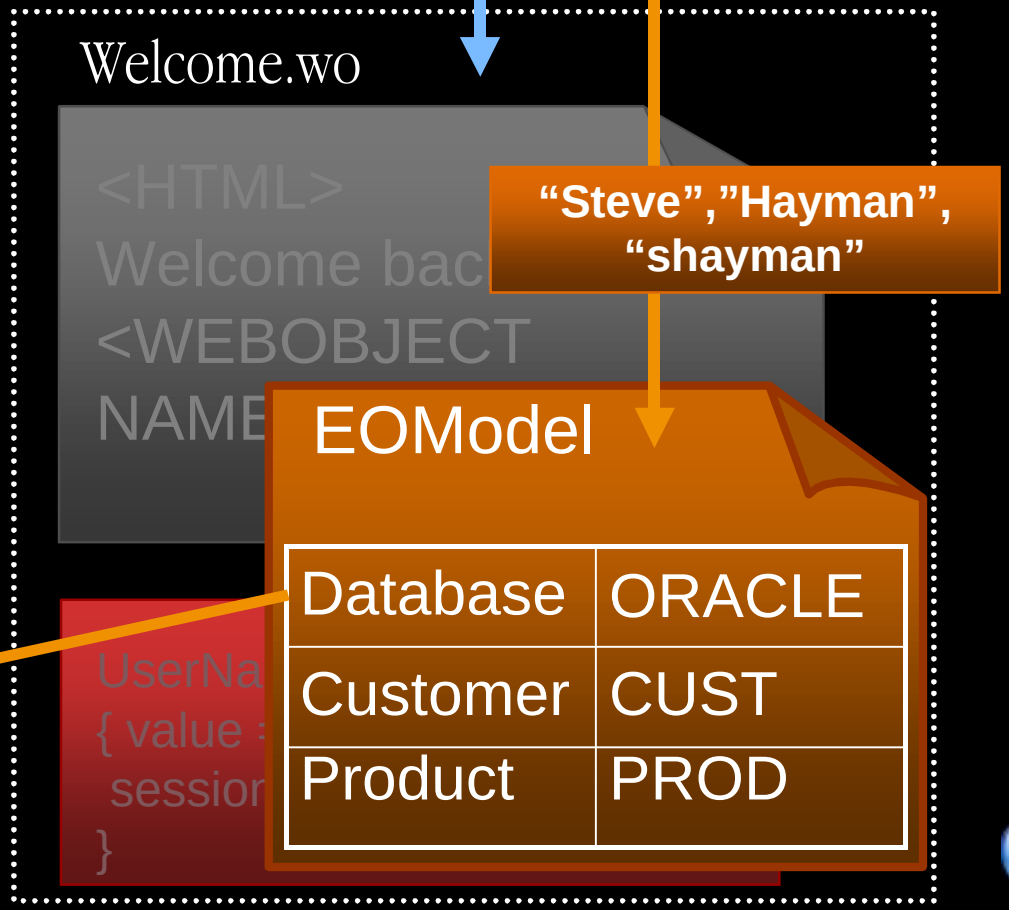


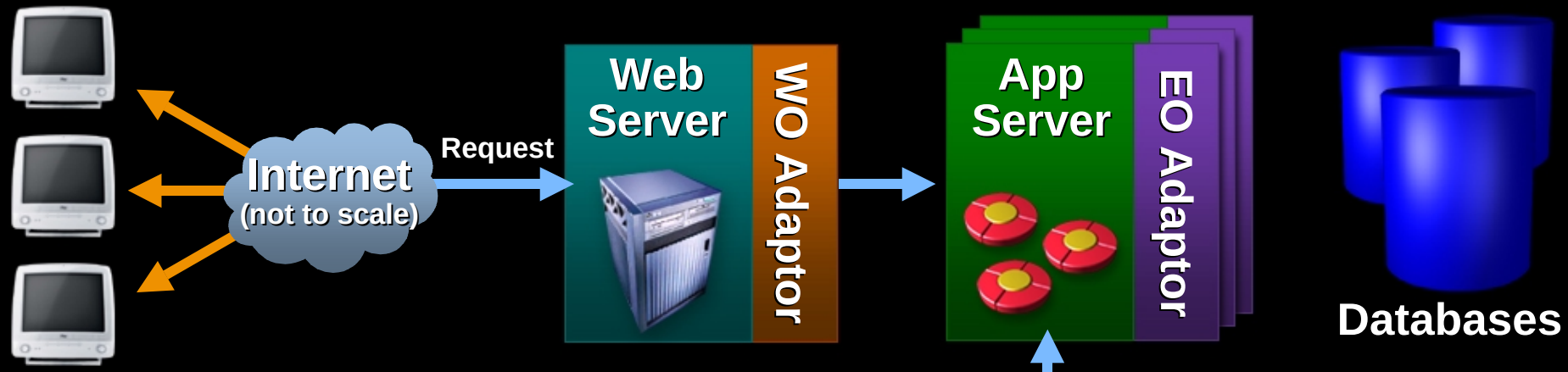
- Any needed objects are fetched from the database
- SQL is generated automatically





- Rows are returned from the database and converted into Enterprise Objects

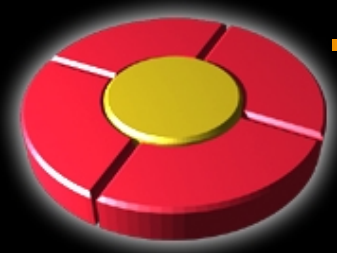




- Messages are sent to EOs

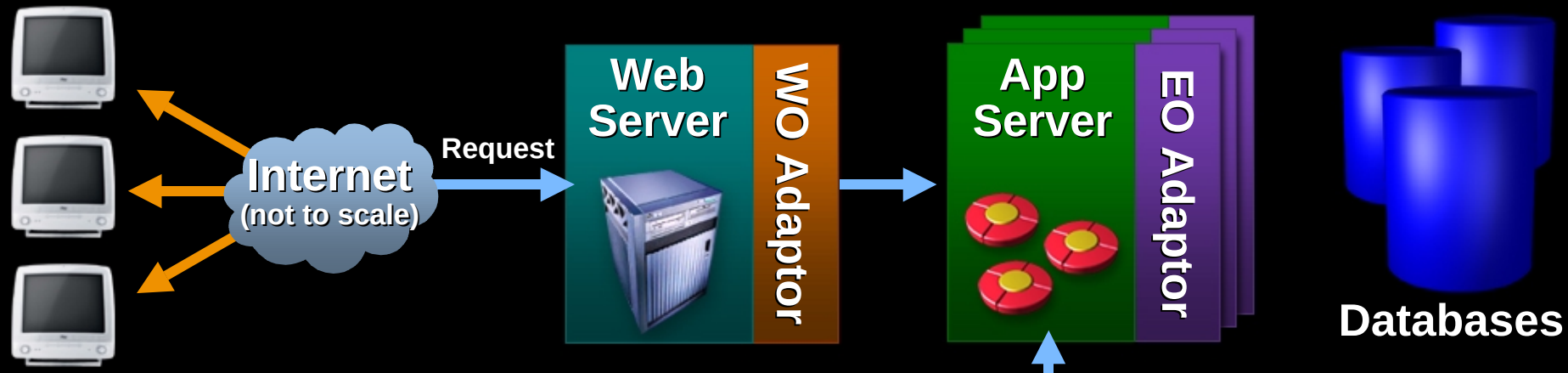
Welcome.wo

```
<HTML>
Welcome back,
<WEBOBJECT
NAME=UserName>
```



"Steve"

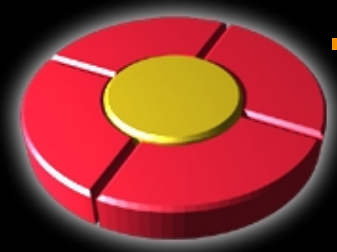
```
UserName: WOString
{ value =
 session.customer.firstName;
}
```



- Response is substituted into HTML template

Welcome.wo

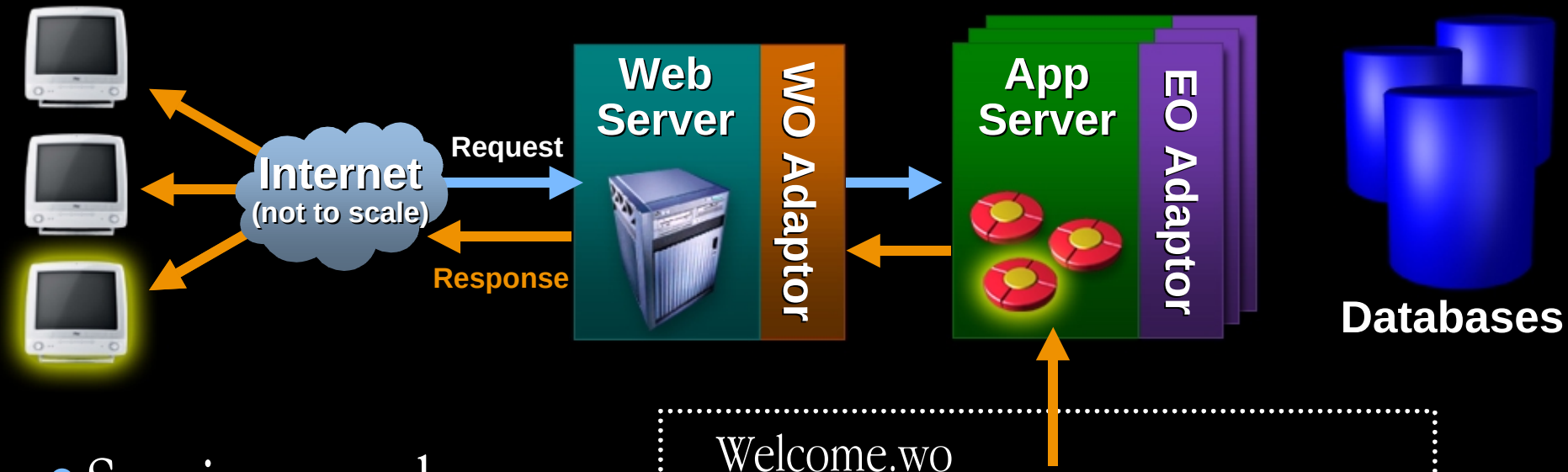
```
<HTML>
Welcome back,
Steve
```



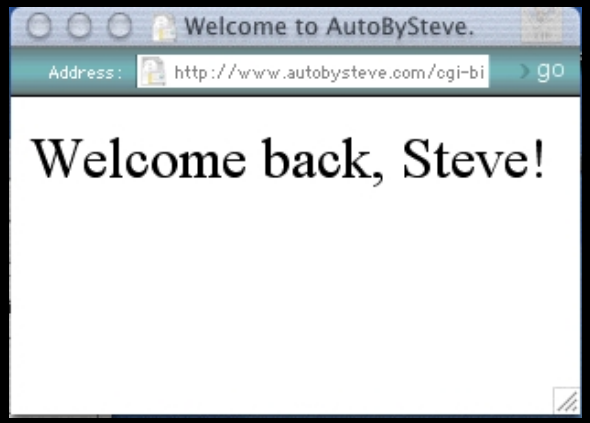
Customer

"Steve"

```
UserName: WOString
{ value =
 session.customer.firstName;
}
```



- Session sends plain HTML back to user



Welcome.wo

```
<HTML>
Welcome back,
Steve
```

```
UserName: WOString
{ value =
 session.customer.firstName;
}
```

- That's the Request/Response Loop

public void **takeValuesFromRequest** (WORequest *aRequest*, WOContext *aContext*)

public WOElement **invokeAction** (WORequest *aRequest*, WOContext *aContext*)

public void **appendToResponse** (WOResponse *aResponse*, WOContext *aContext*)

- Application, Session, and all Components participate (If they want to)

Please Note!

- No mixing of SQL and HTML
- Database, interface, business logic are all completely separate and independent
- No need to write SQL or HTML at all
  (Unless you want to)

The Tools

WebObjects Builder

You'll Create WO Components

- Main.wo/
  - Main.html
  - Main.wod

```
<HTML>
This product
<WEBOBJECT NAME=Image4>
costs
<WEBOBJECT NAME=String1>
<HR>
<WEBOBJECT NAME=Link1>
   Click here
</WEBOBJECT>   to order.
```

```
Image4: WOImage {
   src = "pic.gif"; }
String1: WOString {
  value = product.cost
  numberFormat =
        "$#,###";}
Link1: WOHyperlink {
   action = placeOrder;  }
```

- Main.java

```
public class Order extends WOComponent {
   private ShoppingItem product;
   public WOComponent placeOrder( ) {
     session().addToShoppingCart(product);
     return pageWithName("Checkout");
   }
}
```

You'll Create Lots of WO Components

- One for each "page"
- Subcomponents for headers, footers, navigation bars
- Reusable components for common UI elements
- JavaScript client-side tricks
- Components that contain other components

So How Do You Make Those Components?

- They're just text files
- Use whatever you like
  - emacs
  - vi
  - stickies
  - cat

Oh, Come On

- OK. You can use any WYSIWIG HTML editor
- DreamWeaver
- GoLive
- Or even WebObjects Builder

Inspector

HTML

Objects and actions

WOHyperlink Binding Inspector

Dynamic Inspector

Hyperlink1

Make Static

Attribute	Binding
action	placeOrder
actionClass	
directActionName	
disabled	
fragmentIdentifier	
href	
otherTagString	

Main.wo

None

This product costs

Click here

<BODY> <WOHype

Main

application
session
product

placeOrder

- action
- actionClass
- directActionName
- disabled
- fragmentIdentifier
- href
- otherTagString
- pageName
- queryDictionary
- string
- target

Connect to new binding...

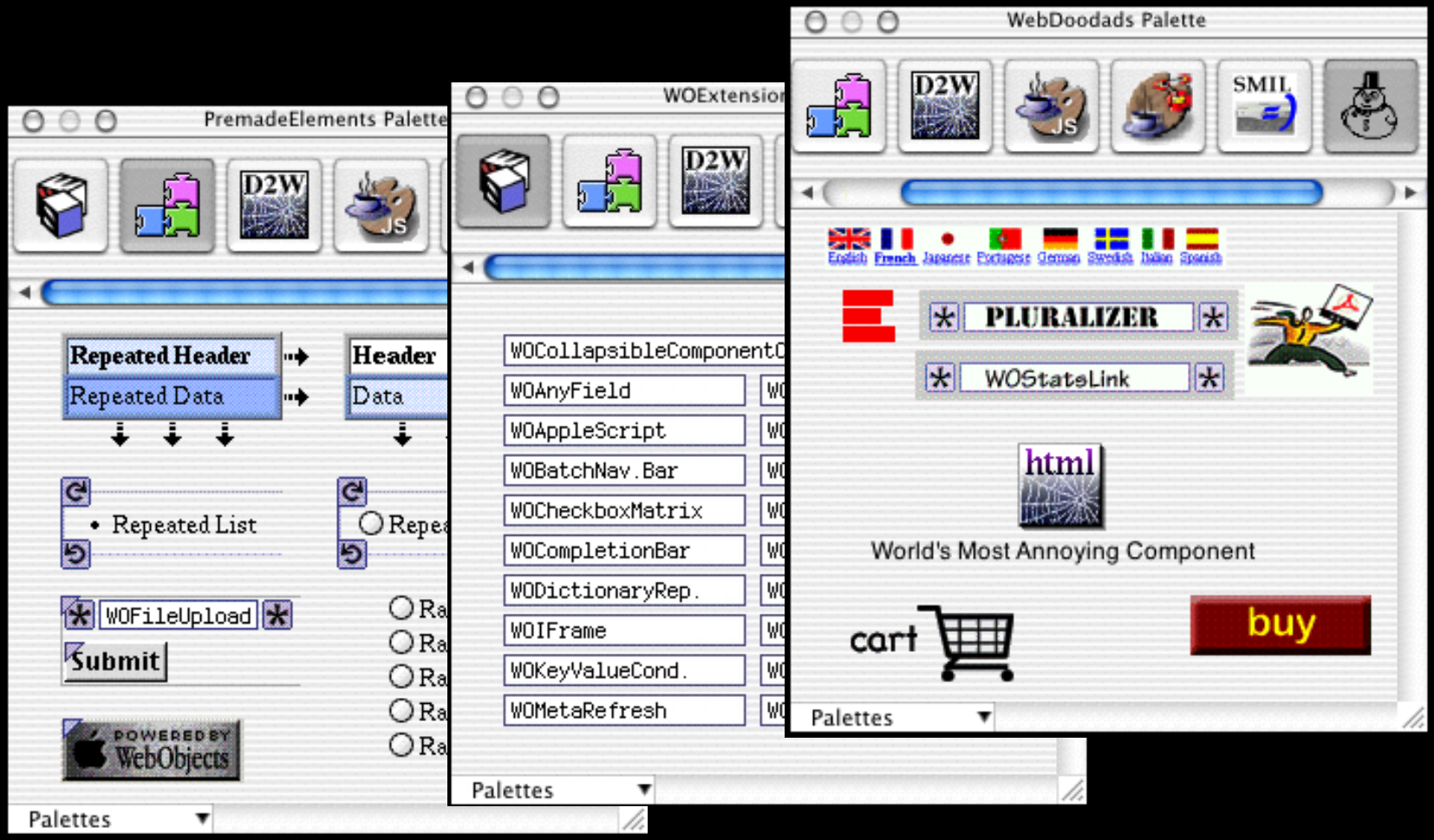
Use WOB in Raw Mode, If You Like

Main.wo

```
<BODY BGCOLOR=#FFFFFF>
    This product
    <WEBOBJECT NAME=Image1></WEBOBJECT>
    costs
    <WEBOBJECT NAME=String1></WEBOBJECT>.
    <WEBOBJECT NAME=Hyperlink1>Click here</WEBOBJECT>
    to order
    <HR></BODY>
```

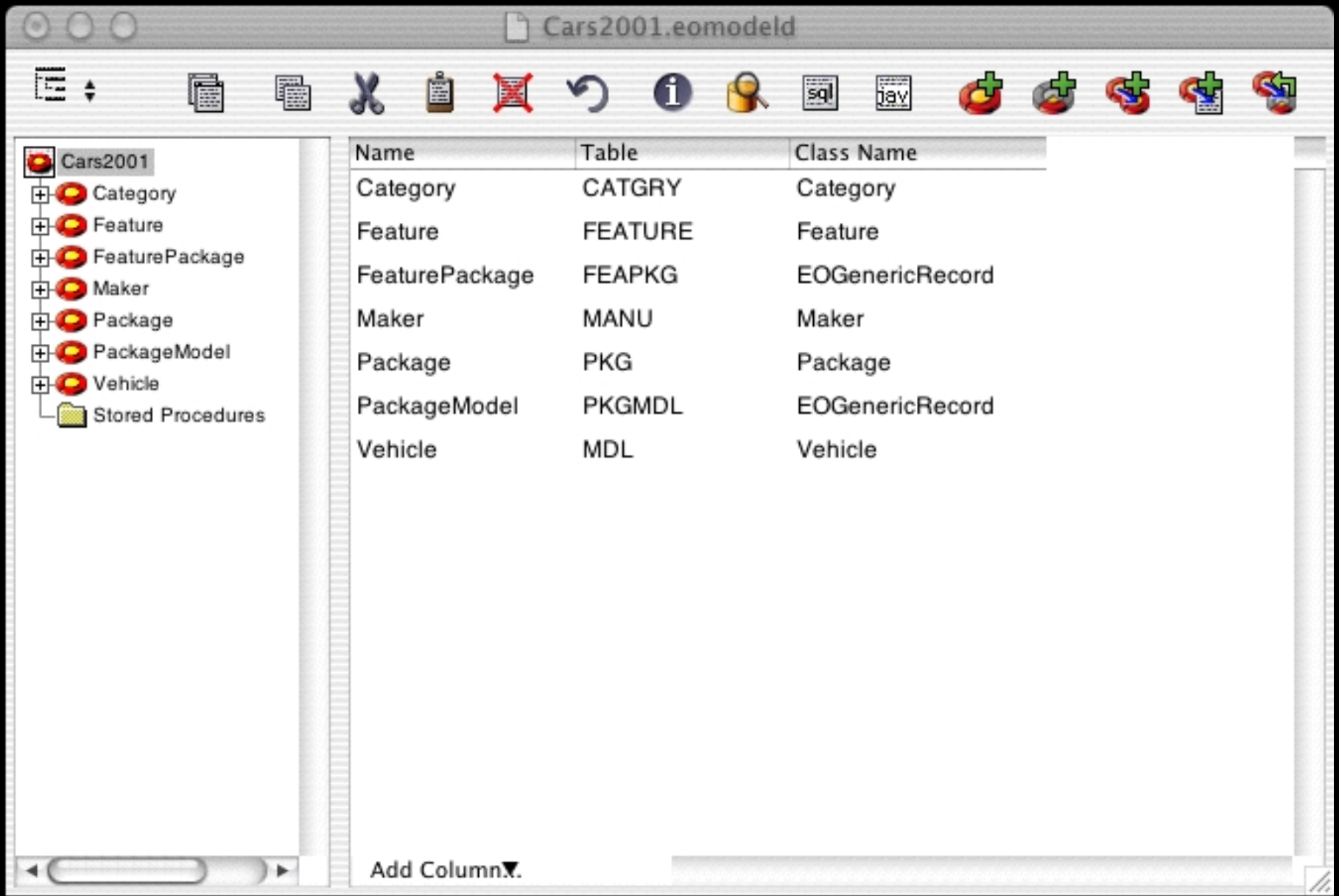
```
Hyperlink1: WOHyperlink {
    action = placeOrder;
}

Image1: WOImage {
}

String1: WOString {
```

Use Palettes to Organize Your Stuff

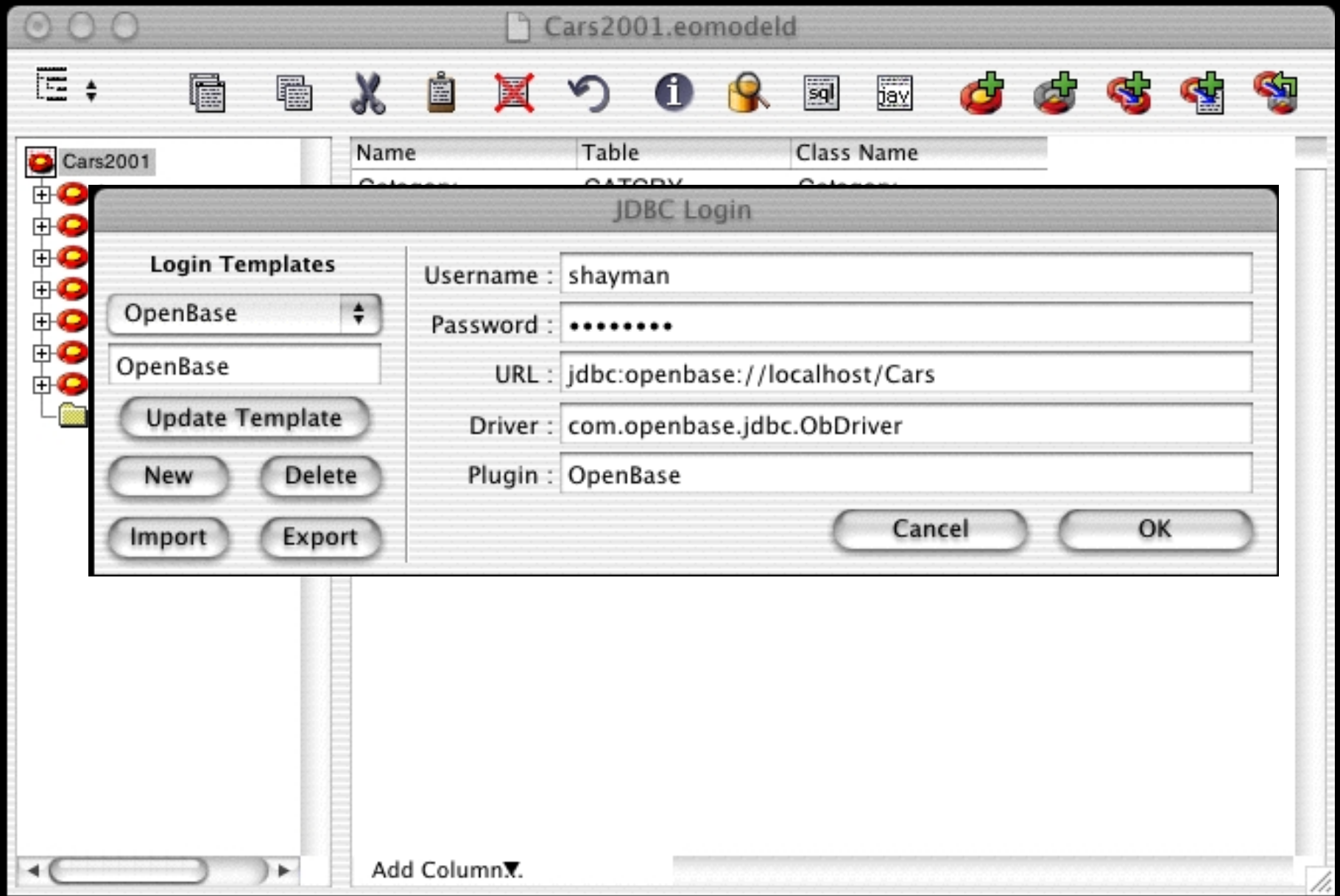


EOModeler

An Editor For *EOModeled* Files



Database Connection Info…



Entities and Tables

Cars2001.eomodeld

- Cars2001
  - Category
  - Feature
  - FeaturePackage
  - Maker
  - Package
  - PackageModel
  - Vehicle
  - Stored Procedures

Name	Table	Class Name
Category	CATGRY	Category
Feature	FEATURE	Feature
FeaturePackage	FEAPKG	EOGenericRecord
Maker	MANU	Maker
Package	PKG	Package
PackageModel	PKGMDL	EOGenericRecord
Vehicle	MDL	Vehicle

Add Column

Vehicle objects <=> the **MDL** table

Entity Information

Cars2001.eomodeld

Cars2001
- Category
- Feature
- FeaturePackage
- Maker
- Package
- PackageModel
- Vehicle
- Stored Procedures

Vehicle Attributes

Name	Column	Value Class	External Type
basePrice	PRMIN	NSNumber	int
blurb	ADTAG	NSString	char
imgJpgData	IMG_JPG	NSData	object
loadedPrice	PRMAX	NSNumber	int
manuid	MANUID	NSNumber	int
mdlname	MDLNAM	NSString	char

Add Column...

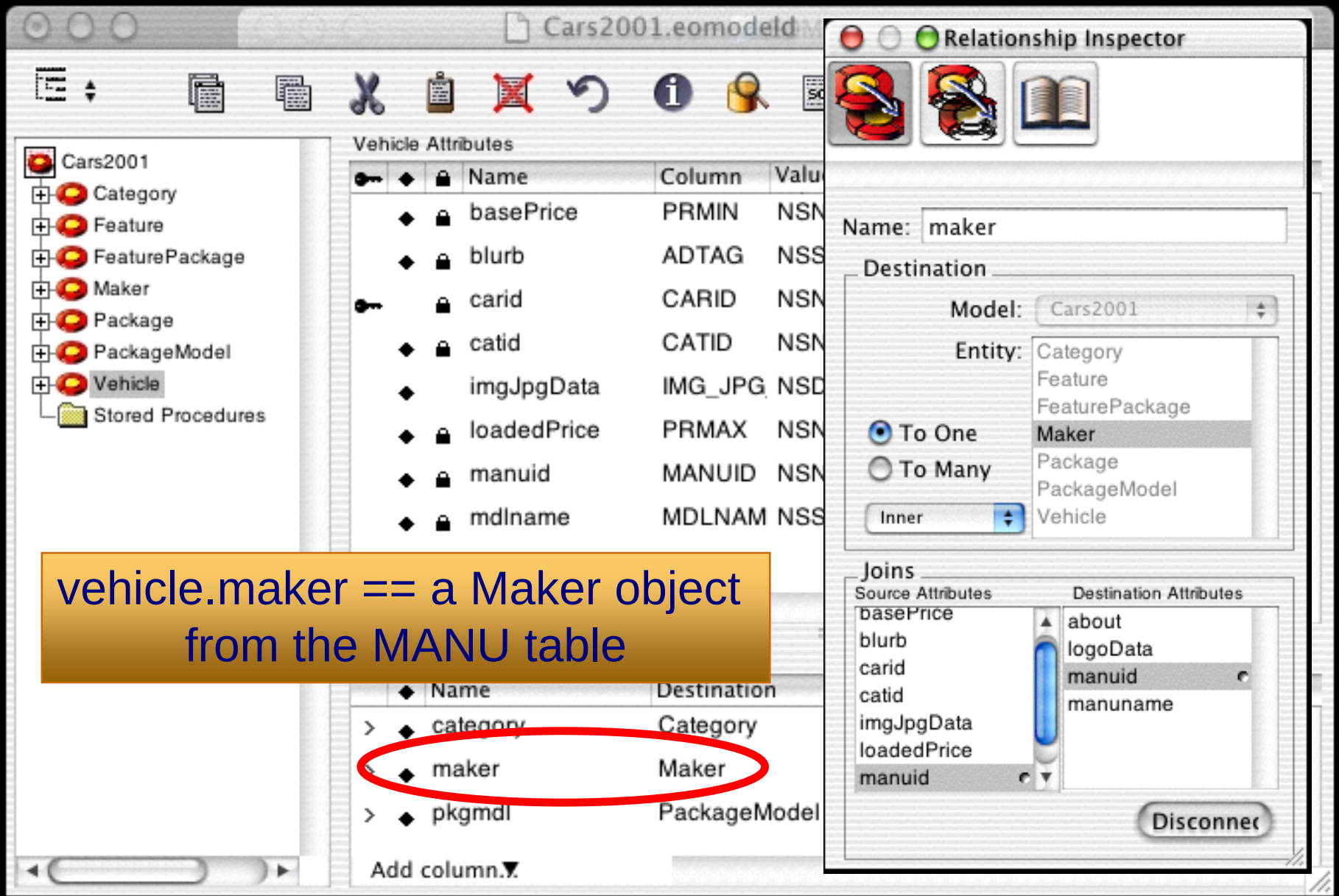
vehicle.basePrice() == the PRMIN column

Vehicle Relationships

Name	Destination	Source Att	Dest Att	Definition
category	Category	catid	catid	
maker	Maker	manuid	manuid	
pkgmdl	PackageModel	carid	carid	

Add column...

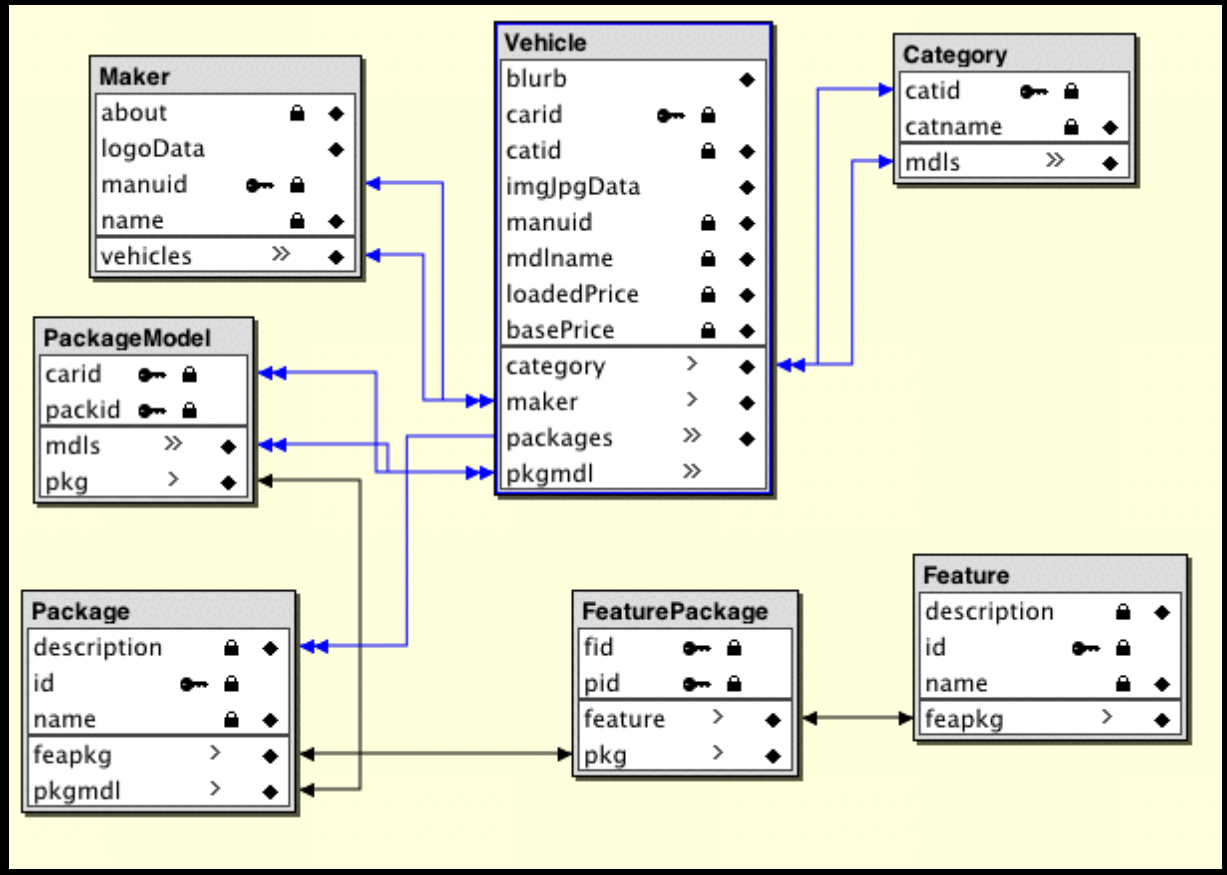
Relationship Information



vehicle.maker == a Maker object from the MANU table

EOModeler

- Model complex relationships between objects too



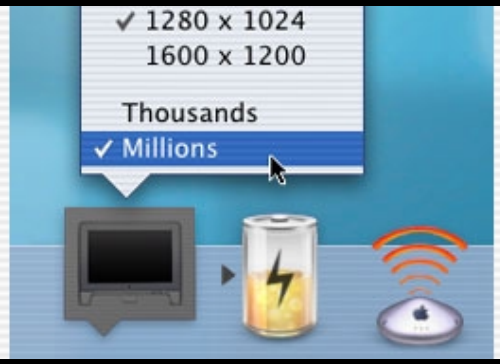
vehicle.maker.name

vehicle.packages

maker.vehicles.@avg.basePrice

Demo

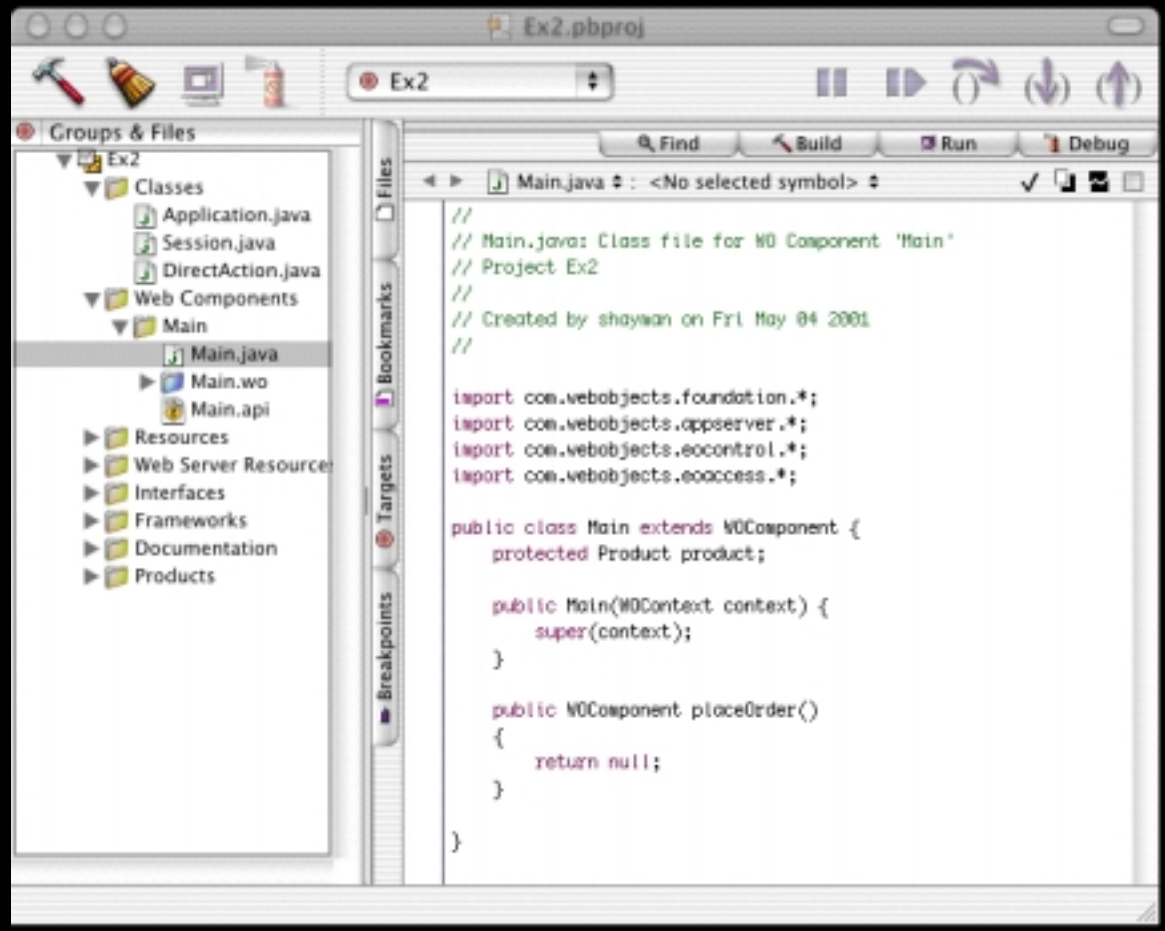
A Really Brief EOModeler Demo



Project Builder

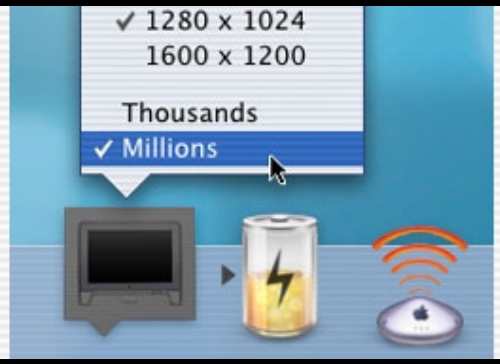
Project Builder

- Coordinates resources
- Code editing and debugging
- Localization



Demo

Now Let’s Build Something



Delivering Other User Interfaces

Nobody said you *had* to use HTML

Other UIs

- Deliver your business objects through other front end technologies
- Java Client
- PDF
- XML
- WAP

Java Client

- A more “traditional” client-server application
- Pure Java application on the client
- WO app on the server
- Enterprise Objects migrate between the two

PDF—High-Quality Printable Documents

- www.reportmill.com
- Object-based PDF generation and reporting
- Subtotaling, grouping, sorting, slicing, dicing

XML

- Turn one or more objects into an XML string with only a line or two of code

```
WOXMLCoder aCoder = WOXMLCoder.coder();
 aString = aCoder.encodeRootObjectForKey(
currentCar, “Vehicle” );
```

```
<Vehicle type="com.apple.yellow.eocontrol.EOGenericRecord" >
  <vehicleName type="java.lang.String">Hummer</vehicleName>
  <basePrice type="java.math.BigDecimal">71000</basePrice>
  ...
</Vehicle>
```

Superduper Assistants

Got a model? You are 95% done.

Direct to Web

Your EOModel

+ Direct to Web

A complete, full featured, and good-looking web app

Direct to Java Client

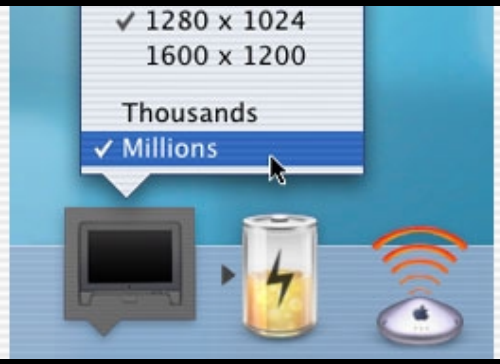
Your EOModel

+ Direct to Java Client

A complete, full featured, and good-looking Java-based application

Demo

Direct to Web
Direct to Java Client



Deploying Your Application

How do you launch your app and keep it running?

Monitor

- A WebObjects application that manages other WebObjects applications
- Tell it what WO Application Server machines you have, and what applications you'd like to run, and how many instances of each
- A service called wotaskd launches them and keeps them running

Monitor in Action

Monitor for WebObjects 5

Back Forward Stop Refresh Home AutoFill Print Mail

Address: http://localhost:52942/cgi-bin/WebObjects/JavaMonitor.woa/wo/N0mb803f9ja80Ixdvf5Aig/21.14.1.37

Monitor

Applications | Hosts | Site | Preferences | Help

AutoBySteve

This page automatically updates every 60 seconds.
Refresh Now

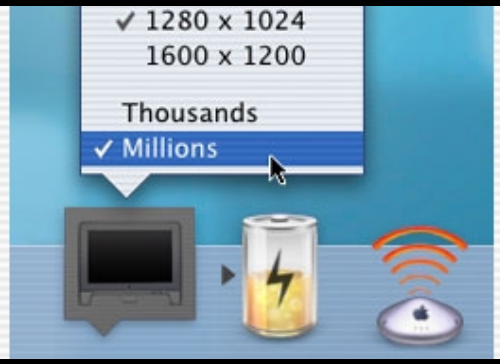
Name	Host - Port	Status	Start - Stop	Auto Recover	Refuse New Sessions	Scheduled	Statistics						Configure	Delete
							Trans-actions	Active Sessions	Average Trans-action	Average Idle Period	Deaths	WOStats		
AutoBySteve-1	yyz2:2001	ON		ON	OFF	OFF	-	-	-	-	-	WOStats	Config	Delete
AutoBySteve-2	yyz2:2002	OFF		ON	OFF	ON	-	-	-	-	-	WOStats	Config	Delete
AutoBySteve-3	yyz2:2003	ON		ON	OFF	OFF	-	-	-	-	-	WOStats	Config	Delete
AutoBySteve-4	yyz2:2004	ON		ON	OFF	ON	-	-	-	-	-	WOStats	Config	Delete
AutoBySteve-5	yyz2:2005	ON		ON	OFF	ON	-	-	-	-	-	WOStats	Config	Delete
ALL INSTANCES:							0	0	0.000	0.000	Clear Deaths	0.00 TPM	Config	

Local machine zone

More on Big Application Deployment

608 Solutions with WebObjects	Civic **Wed., 9:00am**
611 Deploying WebObjects Applications	Room A1 **Thurs., 10:30am**
617 WebObjects Security	Room A1 **Fri., 10:30am**
618 Optimizing WebObjects Applications	Room A1 **Fri., 2:00pm**

What Is the Deal With WebObjects 4.5.1?



WebObjects 4.5.1

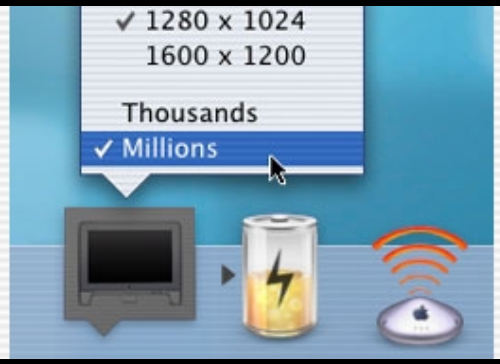
- Mac OS X: Objective-C Only
- Windows 2000, Solaris, HP/UX: Objective-C, Java
  - Complex "Java Bridge" technology to mix ObjC/Java
- Complicated database options
  - Support varies by platform
- End of the road for WO with ObjC.

Doing a NEW project? Use WebObjects 5

Moving From 4.5.1 to 5

- APIs are the same
- JavaConverter
  - Objective-C to Java (mostly)
  - 4.5 Java to 5.0 Java

Learning More



WebObjects Training Classes

- Programming WebObjects I
- Programming WebObjects II
- WebObjects Deployment

http://developer.apple.com/webobjects/schedule.html

WebObjects Lab

- Located downstairs in Room K
- Lab hours
  - Monday 12:00pm–6:00pm
  - Tuesday 10:30am–6:00pm
  - Wednesday 10:30am–6:00pm
  - Thursday 10:30am–6:00pm
  - Friday 10:30am–3:30pm

Vote for WebObjects as Best App Server

http://developer.apple.com/webobjects

Roadmap

603 Introduction to WebObjects Tools	Civic **Tues., 3:30pm**
604 Introduction to Enterprise Objects Framework	Civic **Tues., 5:00pm**
608 Solutions With WebObjects	Civic **Wed., 9:00am**
609 Developing Desktop Applications With WebObjects	Civic **Wed., 10:30am**

All following WebObjects sessions will be in
Room A1

Who to Contact

Toni Trujillo Vian
Director, WebObjects Engineering
webobjects@apple.com

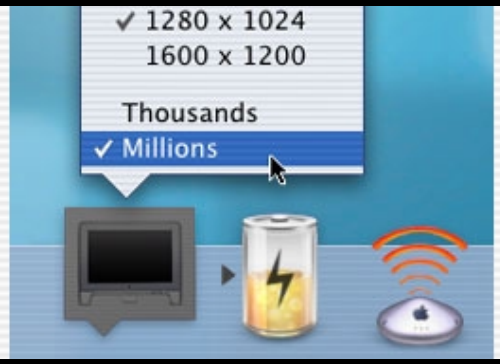
Bob Fraser
WebObjects Product Manager
webobjects@apple.com

iServices Consulting, Integration, Training & Certification
(800) 848-6398
iservices@apple.com

For more information:
www.apple.com/webobjects
developer.apple.com/webobjects
www.apple.com/iservices
www.info.apple.com/webobjects

Subscribe to:
webobjects-announce@apple.com

Q&A



Steve Hayman
Systems Engineer, Apple iServices
iservices@apple.com

http://developer.apple.com/events/urls.html

WWDC

Worldwide Developers Conference 2001